AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A 3D graphics system including:

a producer that outputs graphics commands, said producer comprising a microprocessor;

a consumer that consumes the graphics commands outputted by the producer, said consumer comprising graphics hardware including a cache/command processor and pipelined graphics components that transform and rasterize polygons for display; and

a shared memory coupled to the producer and to the consumer, the shared memory having plural buffers allocated therein, said plural buffers each receiving and temporarily storing graphics commands outputted by the producer for delivery to the consumer, at least some of said commands comprising polygon vertex commands, wherein the producer and the consumer are capable of accessing each said buffer independently of one another,

wherein said buffers store inline commands calling display lists comprising further graphics commands including polygon vertex commands for execution by said graphics hardware, said display lists being stored elsewhere in said shared memory,

wherein said polygon vertex commands are provided to said pipelined graphics components, said pipelined graphics components generating images based at least in part on said polygon vertex commands.

2. (original) A graphics system as in claim 1 wherein the producer and the consumer have independent read and/or write pointers.

3. (Previously Amended) A graphics system as in claim 1 wherein the shared memory stores plural variable sized buffers disposed at selected locations within the shared memory.

4. (Previously Amended) A graphics system as in claim 1 wherein the shared memory stores plural buffers each of which can be independently accessed by the producer and/or the consumer.

5. (original) The graphics system of claim 4 wherein the consumer is incapable of writing to at least an active one of the plural buffers, but maintains, independently of the producer, a write pointer for at least said active one of the plural buffers.

6. (original) The graphics system of claim 2 wherein the consumer selectively increments the consumer write pointer in response to the producer writing to the active buffer.

7. (original) The graphics system of claim 4 wherein the producer provides a producer read pointer and a producer write pointer associated with a first of said plural buffers, and the consumer independently maintains a consumer read pointer and a consumer write pointer associated with said first of said plural buffers.

8. (original) The graphics system of claim 6 wherein the consumer increments the consumer read pointer as the consumer reads from an active buffer, and suspends reading from the active buffer when the incremented consumer read pointer has a predetermined relationship with the consumer write pointer.

9. (Previously Amended) The graphics system of claim 4 wherein a first of the plural buffers includes a read command controlling the consumer to consume the display list which the producer has stored elsewhere within the shared memory, and to resume consuming graphics commands from the first buffer after consuming the display list stored elsewhere.

10. (previously presented) The graphics system of claim 9 wherein the read command specifies a starting address and a length of the display list, the read command controlling the consumer to read the display list of the specified length beginning at the specified starting address.

11. (previously presented) The graphics system of claim 1 wherein at least one of said buffers provides circular first-in-first-out access.

12. (previously presented) The graphics system of claim 1 wherein at least one of said buffers provides first-in-first-out access.

13. (previously presented) The graphics system of claim 1 wherein at least one of the buffers can be selectively attached to both the producer and the consumer simultaneously.

14. (previously presented) The graphics system of claim 1 wherein at least one of the buffers is attached to the producer while a further of said plural buffers is attached to the consumer.

15. (previously presented) The system of claim 1 wherein at least one of the buffers is first attached to the producer, and is subsequently detached from the producer and attached to the consumer.

16. (previously presented) The system of claim 1 wherein at least one of the buffers can be attached to the producer, the consumer, or both.

17. (previously presented) The system of claim 1 wherein only one of said plural buffers is attached to the producer at a time.

18. (previously presented) The system of claim 1 wherein only one of said plural buffers is attached to the consumer at a time.

19. (previously presented) The system of claim 1 wherein each of the buffers has a maximum size of 16 KB.

20. (previously presented) The system of claim 1 wherein the producer sets the size of the buffers.

21. (previously presented) The system of claim 1 wherein at least one of the buffers is dynamically sized to store a frame of graphics commands.

22. (previously presented) The system of claim 1 wherein the producer declares the plural buffers by issuing graphics buffer initialization commands specifying buffer starting address and buffer length.

23. (previously presented) The system of claim 1 wherein at least one of the buffers has a length that is a multiple of 32 bytes and has a minimum size of 64 KB.

24. (previously presented) The graphics system of claim 1 wherein the producer may write a breakpoint into at least one of the buffers, the consumer suspending consumption of graphics commands upon encountering the breakpoint.

25. (previously presented) The system of claim 1 wherein at least one of the buffers has an overflow status indicator indicating when the producer overwrites a location therein.

26. (previously presented) The system of claim 1 further including a hardware status register that indicates the status of at least one of the plural buffers.

27. (previously presented) The system of claim 26 wherein the status register includes the following parameters:

position of a producer write pointer relative to buffer full and buffer empty;

buffer overflow;

whether the producer is currently writing into the at least one buffer; and

whether the consumer is currently reading from the at least one buffer.

28. (previously presented) The system of claim 1 further including a hardware controller coupled to at least one of the plural buffers, the hardware controller providing flow control logic to prevent writes from overrunning reads.

29. (previously presented) The system of claim 1 further including a hardware controller coupled to at least one of the buffers, the hardware controller wrapping read and write pointers from a last location to a first location thereof.

30 (cancelled)

31. (Previously Amended) The system of claim 1 wherein the producer dynamically allocates said buffers within the shared memory.

32. (Currently Amended) A 3D graphics system including:

a memory shared between a producer and a consumer;

a storage buffer allocated within said shared memory, said storage buffer receiving and temporarily storing graphics commands;

a producer that writes graphics commands into said buffer, said producer maintaining a producer write pointer and a producer read pointer associated with the buffer, said producer comprising a microprocessor; and

a consumer that consumes the graphics commands stored within the buffer, said consumer comprising graphics hardware including a cache/command processor and pipelined

graphics components that transform and rasterize polygons for display, the consumer maintaining a consumer write pointer that is independent of the producer write pointer, and a consumer read pointer that is independent of the producer read pointer,

wherein said buffer stores an inline command calling a display command list stored elsewhere in said shared memory for execution by said consumer, said display command list comprising polygon vertex commands that control said pipelined graphics components to draw polygons for display.

33. (original) The graphics system of claim 32 wherein the consumer increments the consumer read pointer each time the consumer consumes from the buffer, and suspends consumption from the buffer when the consumer read pointer has a predetermined relationship with the consumer write pointer.

34. (original) The graphics system of claim 32 wherein the consumer selectively auto-increments the consumer write pointer in response to the producer writing to the buffer.

35. (previously presented) A graphics system including:

a storage buffer that receives and temporarily stores graphics commands;

a producer that writes graphics commands into said buffer, said producer maintaining a producer write pointer and a producer read pointer associated with the buffer; and

a consumer that consumes the graphics commands stored within the buffer, the consumer maintaining a consumer write pointer that is independent of the producer write pointer, and a consumer read pointer that is independent of the producer read pointer,

wherein the producer sends the consumer a configuration command specifying whether the consumer should auto-increment the consumer write pointer in response to producer writes to the buffer.

36. (Currently Amended) In a 3D graphics system including a graphics command producer that writes graphics commands into a buffer based on a producer write pointer, said command producer allocating said buffer within a memory shared by said command producer and a graphics command consumer, the graphics command consumer reading graphics commands from the buffer based on a consumer read pointer, said graphics command consumer comprising graphics hardware including a cache/command processor and pipelined graphics components that transform and rasterize polygons for display,

an improvement comprising:

a consumer write pointer independently maintained by the consumer, the consumer write pointer indicating the extent of valid data the producer has written into said buffer, the consumer ceasing to consume graphics commands from the buffer upon the consumer read pointer having a predetermined relationship to the consumer write pointer,

wherein said buffer stores an inline command calling a list of graphics commands stored elsewhere in said shared memory for execution by said consumer, said called list of graphics commands including polygon vertex commands that command said pipelined graphics components to draw polygons for display.

37. (Currently Amended) In an interactive 3D graphics system including a processor module executing an application, a graphics processor module and at least one memory coupled

to the processor module and to the graphics processor module, a method of controlling the flow of graphics commands between the processor module and the graphics processor module comprising:

dynamically establishing, under control of the application, a variable number of FIFO buffers in the memory, the application specifying the size of each of the FIFO buffers;

the application controlling the processor module to write graphics commands into at least a first of the plurality of FIFO buffers; and

the application sending graphics commands to the graphics processor module that control the graphics processor module to read, independently of said processor writes, the graphics command from the first FIFO buffer,

wherein said application stores, into at least one of the FIFO buffers, an inline command calling a list of graphics commands stored elsewhere in said memory for execution by said graphics module, said called list of graphics commands including polygon vertex commands that command said graphics module to draw polygons for display.

.

38. (original) The graphics system of claim 37 wherein the processor module provides a processor module read pointer and a processor module write pointer associated with a first of said plurality of buffers, and the graphics processor module independently maintains a graphics processor module read pointer and a graphics processor module write pointer associated with said first buffer.

39. (original) The graphics system of claim 37 wherein the graphics processor module increments the graphics processor module read pointer each time the graphics processor module

reads from the first buffer, and suspends reading from the first buffer when the graphics processor module read pointer has a predetermined relationship with the graphics processor module write pointer.

40. (original) The graphics system of claim 37 wherein the graphics processor module selectively auto-increments the graphics processor module write pointer in response to the processor module writing to the first buffer.

41. (original) The graphics system of claim 37 wherein the graphics processor module maintains, independently of the processor module, a write pointer for at least an active one of the plurality of buffers.

42. (original) The method of claim 37 further including setting a breakpoint, and at least temporarily suspending the graphics processor module from reading a buffer in response to the graphics processor module encountering the breakpoint.

43. (original) The method of claim 37 wherein the selectively controlling step includes suspending the processor module from writing to the buffer upon detection of an overflow.

44-54 (cancelled)

55. (Currently Amended) A 3D graphics system including:

a shared memory that receives and temporarily stores graphics commands;

a producer that writes commands into a buffer within said shared memory, said producer comprising a microprocessor, said commands including a first set of graphics commands and a command referencing a second set of graphics commands stored elsewhere within said shared memory; and

a consumer that consumes the first set of graphics commands stored within the buffer and, in response to encountering the referencing command, consumes the second set of graphics commands referenced thereby and subsequently returns to the buffer to consume additional commands therefrom, said consumer comprising graphics hardware including a cache/command processor and further graphics pipeline components that transform and rasterize polygons for display,

wherein said buffer stores an inline command calling a list of display commands stored elsewhere in said shared memory for execution by said consumer, said called list of graphics commands including polygon vertex commands that command said graphics pipeline components to draw polygons for display.

56. (original) The graphics system of claim 55 wherein the buffer is a circular buffer.

57. (previously presented) A graphics system including:

a storage device that receives and temporarily stores graphics commands;

a producer that writes commands into a buffer within said storage device, said commands including a first set of graphics commands and a command referencing a second set of graphics commands stored elsewhere within said storage device; and

a consumer that consumes the first set of graphics commands stored within the buffer and, in response to encountering the referencing command, consumes the second set of graphics commands referenced thereby and subsequently returns to the buffer to consume additional commands therefrom,

wherein the referencing command specifies a starting address of a display list, the referencing command controlling the consumer to read the display list beginning at the specified starting address.

58. (previously presented) A 3D graphics system including:

a storage device that receives and temporarily stores graphics commands;

a producer that writes commands into a buffer within said storage device, said commands including a first set of graphics commands and a command referencing a second set of graphics commands stored elsewhere within said storage device; and

a consumer that consumes the first set of graphics commands stored within the buffer and, in response to encountering the referencing command, consumes the second set of graphics commands referenced thereby and subsequently returns to the buffer to consume additional commands therefrom,

wherein the referencing command specifies a number of data units the consumer is to consume.

59. (original) The graphics system of claim 55 wherein the consumer is incapable of writing to the buffer, but maintains, independently of the producer, a write pointer for the buffer.

60. (Currently Amended) In a 3D graphics system, a method for passing graphics commands from a producer of graphics commands, said producer comprising a microprocessor, to a consumer of graphics commands, said consumer comprising graphics hardware including a cache/command processor and further graphics pipeline components that transform and rasterize polygons for display, the method comprising:

creating plural variable sized buffers disposed at variable locations within a memory coupled to the producer and the consumer,

temporarily storing graphics commands produced by the producer in the variable sized buffers;

writing, into at least one said buffers, an inline command calling a list of display commands stored elsewhere in the memory;

consuming the graphics commands from the variable sized buffers with the consumer by accessing the buffers independently of the producer;

executing the list of display commands in response to said inline command within said at least one buffer, said list of display commands including polygon vertex commands specifying vertices of polygons to draw within a graphics image;

after consuming the list of display commands, accessing further graphics command within said buffer having said inline command calling said display list written therein; and

generating at least a part of ~~a~~ said graphics image based at least in part on the consumed graphics commands and said executed display list.

61. (original) The method of claim 60 wherein the consumer is incapable of writing to at least an active one of the plural buffers, and the method further includes the consumer

maintaining, independently of the producer, a write pointer for at least said active one of the plural buffers.

62. (original) The method of claim 60 further including maintaining, with the producer, a producer read pointer and a producer write pointer associated with a first of said plural buffers, and the independently maintaining, with the consumer, a consumer read pointer and a consumer write pointer associated with said first of said plural buffers.

63. (original) The graphics system of claim 62 further including incrementing, with the consumer, a consumer read pointer as the consumer reads from an active buffer, and suspending reading from the active buffer when the incremented consumer read pointer has a predetermined relationship with the consumer write pointer.

64. (original) The graphics system of claim 63 further including selectively incrementing the consumer write pointer in response to the producer writing to the active buffer.

65. (previously presented) The method of claim 60 wherein a first of the plural buffers includes a read command, and the method further includes:

(a) consuming the display list the producer has stored elsewhere within the memory in response to encountering the read command, and

(b) resuming consumption of graphics commands from the first buffer after consuming the display list stored elsewhere.

66. (previously presented) The graphics system of claim 65 wherein the read command specifies a starting address and a length of the display list, and step (a) includes controlling the consumer to read the display list of the specified length beginning at the specified starting address.

67. (original) The method of claim 60 wherein a first of the plural buffers provides circular first-in-first-out access.

68. (previously presented) The method of claim 60 wherein a first of the plural buffers provides first-in-first-out access.

69. (original) The method of claim 60 further including selectively attaching any of the plural buffers to both the producer and the consumer simultaneously.

70. (original) The method of claim 60 further including attaching a first of the plural buffers to the producer and attaching a second of the plural buffers to the consumer.

71. (original) The system of claim 60 further including attaching a first of the plural buffers to the producer, and subsequently detaching the first buffer from the producer and attaching the first buffer to the consumer.

72. (original) The system of claim 60 further including attaching any of the plural buffers to the producer, the consumer, or both.

73. (original) The system of claim 60 further including attaching only one of the plural buffers to the producer at a time.

74. (original) The system of claim 60 further including attaching only one of the plural buffers to the consumer at a time.

75. (Currently Amended) A method for producing 3D images including:

maintaining a producer write pointer and a producer read pointer associated with a buffer allocated within a memory shared by the producer and a consumer;

writing graphics commands, including an inline display list call, into the buffer, and updating at least the write pointer in response to the writing;

maintaining, in association with the buffer, a consumer write pointer that is independent of the producer write pointer, and a consumer read pointer that is independent of the producer read pointer;

the consumer consuming the graphics commands stored within the buffer, including accessing a display list comprising polygon vertex graphics commands stored elsewhere within said memory in response to encountering said inline display list call within said buffer, updating at least the read pointer in response to the consuming, and, after executing the display list, accessing and consuming further graphics command disposed sequentially within said buffer after the inline display list call; and

producing at least a part of a graphics image at least in part in response to the consuming and executing step.

76. (Currently Amended) A method of producing 3D images including:

writing commands into a buffer within a shared memory, said commands including a first set of graphics commands and a command referring to a display list stored elsewhere within said shared memory;

consuming the first set of graphics commands stored within the buffer;

in response to encountering the referring command, executing polygon vertex commands in the display list and subsequently automatically returning to consume additional commands from the buffer; and

generating at least a part of an image at least in part in response to the consumed first set of graphics commands and said display list.

77-80 (cancelled)

81. (Currently Amended) A method of supplying 3D graphics commands to a 3D graphics command consumer comprising:

(a) storing a command sequence beginning at a predetermined storage location;

(b) supplying a graphics command stream through a FIFO buffer to the consumer, the stream including at least one command that refers the command consumer to a display list comprising polygon vertex commands stored beginning at the predetermined storage location; and

(c) the graphics command consumer executing said display list polygon vertex commands beginning at the predetermined storage location in response to encountering said at least one command that refers the command consumer to the display list,

wherein the producer returns to access additional portions of said graphics command stream from the FIFO buffer after consuming the command sequence beginning at the predetermined storage location.

82 (presently presented). A method of efficiently generating successive graphics images on a display device, comprising:

writing commands into a memory buffer, said commands including at least a first set of graphics commands and at least one calling command that calls a further, prestored list of display commands;

consuming, with a 3D graphics engine, at least some of the first set of graphics commands to generate at least a portion of a first image in a frame buffer memory;

in response to encountering the calling command, reading the stored display commands in the display list with the 3D graphics engine and responsively generating at least a further portion of said first image in said frame buffer memory;

the 3D graphics engine subsequently automatically returning from said called display list to consume additional ones of the first set of graphics commands stored in said memory buffer to generate at least an additional portion of the first graphics display image in the frame buffer memory;

displaying the first graphics display image on the display device;

writing additional commands into the same or different memory buffer, said additional commands including at least a second set of graphics commands and at least one calling command that calls the same further, prestored list of display commands, ;

consuming, with the 3D graphics engine, at least some of the second set of graphics commands to generate at least a portion of a second image in the same or different frame buffer memory;

in response to encountering the calling command, reading the commands in said prestored display list with the 3D graphics engine and responsively generating at least a further portion of said second image in said same or different frame buffer memory, thereby reusing said further, stored list of display commands to generate said second image;

subsequently automatically returning from said called display list to consume, with the 3D graphics engine, additional ones of the second set of graphics commands stored in said memory buffer to generate at least an additional portion of the second graphics display image in the same or different frame buffer memory; and

displaying the second graphics display image on the display device based on the contents of said same or different frame buffer memory.

83 (presently presented). The method of claim 82 wherein said consuming comprises consuming in a first-in-first-out order, thereby relaxing the degree to which the 3D graphics engine needs to be synchronized to the graphics command producer.

84 (presently presented). The method of claim 82 further including allocating a variable number of variable sized memory buffers for receiving and storing graphics command sets.